Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 28 (2014) 997–1017

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Fast vocabulary acquisition in an NMF-based self-learning vocal user interface

Bart Ons *, Jort F. Gemmeke, Hugo Van hamme

Department ESAT-PSI, KU Leuven, Leuven, Belgium

Received 11 April 2013; received in revised form 7 March 2014; accepted 14 March 2014
Available online 25 March 2014

Abstract

In command-and-control applications, a vocal user interface (VUI) is useful for handsfree control of various devices, especially for people with a physical disability. The spoken utterances are usually restricted to a predefined list of phrases or to a restricted grammar, and the acoustic models work well for normal speech. While some state-of-the-art methods allow for user adaptation of the predefined acoustic models and lexicons, we pursue a fully adaptive VUI by learning both vocabulary and acoustics directly from interaction examples. A learning curve usually has a steep rise in the beginning and an asymptotic ceiling at the end. To limit tutoring time and to guarantee good performance in the long run, the word learning rate of the VUI should be fast and the learning curve should level off at a high accuracy. In order to deal with these performance indicators, we propose a multi-level VUI architecture and we investigate the effectiveness of alternative processing schemes. In the low-level layer, we explore the use of MIDA features (Mutual Information Discrimination Analysis) against conventional MFCC features. In the mid-level layer, we enhance the acoustic representation by means of phone posteriorgrams and clustering procedures. In the high-level layer, we use the NMF (Non-negative Matrix Factorization) procedure which has been demonstrated to be an effective approach for word learning. We evaluate and discuss the performance and the feasibility of our approach in a realistic experimental setting of the VUI-user learning context.

Keywords: MIDA; Phone posteriorgram; NMF; Fast learning; Vocabulary acquisition

1. Introduction

Command-and-control (C&C) speech recognition allows users to interact with systems like domestic devices, assistive technology, computers, smart-phones or other mobile devices. The user speaks a command or a phrase to control different functions in the environment like the central heating or the light units in the house, to retrieve information on their smartphone or to navigate through a menu on a computer. C&C applications are especially useful

* Corresponding author. Tel.: +32 16321071.
E-mail address: Bart.ons@esat.kuleuven.be (B. Ons).

http://dx.doi.org/10.1016/j.csl.2014.03.004

for people with a physical disability affording them handsfree control of their wheel chair, the positioning of their bed or other independent living aids.

In most speech driven C&C applications, the spoken commands are restricted to a predefined list of phrases described by a restricted grammar and vocabulary. The size of the vocabulary ranges from a few to a few hundred words and the grammars are mainly rule-based. Although the targeted VUI application allows a developer to consider many interaction scenarios beforehand, the use of a VUI is not always successful when the interaction oversteps the clear boundaries of the lexicon, the grammars or the dialogue models. Even in less restrictive frameworks, such as in the now popular Siri speech recognition application for the iPhone, performance degrades rapidly if the acoustic models do not match the speech material used to train the system, for example on accented or dysarthric speech. The goal of this paper is to investigate a VUI model which is able to associate any utterance to a C&C action allowing command and control usability by deviant speech as well.

Over the past decade, various approaches have been proposed for adaptation to unexpected circumstances in real-life situations. For instance, Paek and Chickering (2007) proposed a statistical model for mobile devices that tracks the past of the user's behaviour in order to predict commands. In Heinroth et al. (2012), grammars were able to adapt dynamically to real-life communication making interactions more natural. In Potamianos and Narayanan (1998) and Kuhn et al. (2000), speaker-independent acoustic models were adapted to speaker-dependent models allowing for better recognition of the user-specific vocalizations. There are plenty more studies that paved the way to more natural interaction with machines and devices by means of human-centred design and user adaptation. For instance, in a study of Parker et al. (2006) a robust speech recogniser was developed to adapt to dysarthric speech as well. In the "Speech Training And Recognition for Dysarthric Users of Assistive Technology" (STARDUST) project (Parker et al., 2006), the problem was tackled by adaptation in two directions: a training package assisting dysarthric speakers to improve the recognition likelihood of their utterances (users adapting to speech recognition systems) and speech recognition systems having greater tolerance to variability of dysarthric vocalizations (speech recognition models adapting to users) were developed.

However, all these approaches have in common that these systems are still based on acoustic and language models that are trained beforehand and adapted through interaction to the spoken utterances of the user. While these methods focus on adaptation, we focus on *grounding*: learning both vocabulary and acoustics directly from the user during the usage of the VUI. The grounding process (Clark and Schaefer, 1989) refers to the process by which common ground or meaning is built between the user and the system. Situated in the "Adaptation and Learning for Assistive Domestic Vocal Interfaces" (ALADIN) project (van de Loo et al., 2012; Gemmeke et al., 2013), we aim to design a VUI that learns to understand speech by mining the speech input from the end user and the changes that are provoked on a device.

The VUI should learn to understand classes referring to devices, actions or properties by using cross-situational evidence and learning the statistical regularities between two modalities, namely, the spoken utterances of the user and the feedback coming from the device(s). Supervision coming from the device is weak in the sense that the information provided to the VUI consists of signals referring to states and actions in a machine without any chronological information, orthographic nor phonetic transcriptions. Earlier studies have demonstrated that multi-modal Non-negative Matrix Factorisation (NMF) is a useful tool to learn weakly co-occurring regularities over two modalities in order to find the intra- and inter-modality patterns. For instance, in Caicedo et al. (2012), NMF is used to generate multimodal image representations that integrate visual and text features for image collections guided by ratings, comments and tags on the web. Akata et al. (2011) used a similar approach and called it multiview clustering to cluster images and predict image labels. Similar to NMF-based keyword discovery in Driesen et al. (2012a), we use NMF to learn co-occurrences between acoustic feature vectors emerging from the spoken utterances and semantic label vectors describing the action properties.

In order for a self-learning approach to be useful as a VUI, the learning process should be *fast*. At the same time, after sufficient training tokens have been presented, the *accuracy* should be high. The contribution of this work is twofold. First, we investigate to what extent the learning speed and accuracy can be improved by using more advanced feature representations in NMF. We use phone classifiers to create phone confidence measures to replace the conventional acoustic input in NMF learning (Driesen, 2012; Sun, 2012). In addition to phone classifiers, we also evaluate a speaker-dependent version of soft Vector Quantization (soft VQ), which is a data-driven and probabilistic procedure to cluster the acoustic data of the speaker. We tested the usefulness of this data-driven approach for small training sets as user-specific data is expected to be scarce in the beginning of the VUI usage.

A second contribution of this work is that we investigate to what extent the NMF machine learning procedure can be used for a VUI under realistic constraints. While previously, NMF evaluations were typically speaker-independent, we will work speaker-dependent since the system is self-learning and builds its representations from scratch. Also, while there is some prior work on investigating the learning speed of NMF (ten Bosch et al., 2009; Driesen and Van hamme, 2012a; Ons et al., 2012), this made unrealistic assumptions on the amount of speech material available during training for building the lower-level acoustic representations. Here, we will use speaker-independent material from different annotated datasets to train the phone classifiers or VQ clusters beforehand and use only speaker-dependent training data in proportion to the expected accumulative production of speech in a real VUI-user learning context, therefore, evaluating the feasibility of the VUI by limiting access to available data corresponding to a realistic operating mode.

The remainder of the paper is organised as follows. In Sections 2 and 3, we introduce the learning framework, including the feature representations, acoustic models and NMF procedure used throughout the paper. In Section 4, we conduct a series of experiments to evaluate the effectiveness of the NMF approach on the ACORNS database (Boves et al., 2007) containing normal speech. NMF learning has been evaluated on ACORNS data (e.g. Driesen, 2012) and therefore we use ACORNS as well to introduce a proper baseline. We discuss related work and present our thoughts on future work in Section 5.

2. A self-taught user interface

2.1. The learning problem

Self-learning refers to the VUI's ability to learn from interactions with the end user. The training token consists of speech paired with the demonstration of the intended action. For instance, the user utters the command: "Close the door, please" and the VUI forwards that command to the automatic door closing system. However, if the VUI is lacking confidence, the user is asked to demonstrate the intended action, for instance, by pushing the correct button on an environmental control system (EVS). The VUI inferres the executed action from information sent by the control device. This assumes that a number of properties and actions enabling the control of a device are predetermined and a placeholder is provided for each one of them to represent the spoken words and to relay them to this control information during the learning process. The user's command and the demonstrated action is counting as one training example. If the VUI parses the wrong command, then the user has the opportunity to overrule the action. The overruling action will then serve as grounding information. In this study, we evaluate how many demonstrated actions, i.e. a command and a correct demonstration, are necessary to obtain a particular performance.

In Table 1, the learning problem is demonstrated by means of a toy example. Supervision in Table 1(a) is displayed by the pictograms representing semantic tags like a device, an activity, or a property produced by a button-push. Assuming that the acoustic representations of the spoken utterances are represented by the text characters and the semantic tags by the pictograms in respectively the first and the second column of Table 1(a), then the learning process consists of finding the recurrent acoustic patterns (at least two adjacent letters) and their co-occurring semantic tags that make up the discriminative parts of the user's commands. These recurring patterns are displayed in Table 1(b).

2.2. Non-negative Matrix Factorization

Learning a word by means of the statistical co-occurrence of multimodal evidence across situations is called cross-situational learning (Quine, 1964). In earlier studies (ten Bosch et al., 2009; Van hamme, 2008; Driesen et al., 2012a), weakly supervised Non-negative Matrix Factorization (NMF) has been presented as a useful machine learning procedure to discover and learn the acoustic representation of words accompanied by weak supervision. NMF works by factorizing a collection of utterance-based representations into the product of a matrix containing the latent factors describing the recurrent acoustic patterns (such as words) in utterances, and a matrix describing for each utterance which latent factors are active. In weakly supervised NMF, the utterance-based representations are accompanied by grounding information, i.e. label vectors referring to the semantic tags, and the aim is to find the recurrent patterns that co-occur with the semantic tags in the first matrix of the factorization and the activations of these semantic tags in the second matrix component (see Table 1).

Table 1
The learning problem. The letters in the top table represent the acoustic signal, italic text indicates a recurrent pattern and the bold text represents the co-occurrence with the semantic tags that are displayed in the second column. From the cross-situational evidence, the acoustic feature representation for each semantic tag as displayed in the bottom table should be learned.

(a) VUI training samples

Phrases	Tags
look, *I'm* ***closing*** *the* ***window***	▶◀ ⊞
door close, please	▶◀ ⊡
I'm ***opening*** *the* ***door*** *now*	◀▶ ⊡
open *the* ***window***	◀▶ ⊞
switch on *the* ***TV***	○ 📺
the ***heating*** system should be ***turned*** ***off***	— ▥
I ***turn off*** *the* ***TV***	— 📺
I ***switch on*** *the* ***heating***, *now*	○ ▥

(b) Associations with semantic tags

▶◀	⊞	◀▶	⊡	—	📺	○	▥
c	w	o	d	t	T	s	h
l	i	p	o	u	V	w	e
o	n	e	o	r		i	a
s	d	n	r	n		t	t
	o			o		c	i
	w			f		h	n
				f		o	g
						n	

3. Architecture

3.1. Overview

In this study, we investigate whether different preparations of the data lead to different learning rates in NMF. We consider two types of training sets. The first type is called the *keyword-learning training set* and it contains the so-called correctly supervised learning examples occurring in our simulated VUI-user usage context. The keyword-learning training set is used to build the NMF model, i.e. latent acoustic representations for C&C keywords, and it is based on the ACORNS database. However, some steps in the processing flow use acoustic models to prepare the feature vectors for NMF and these models require training too. For instance, a phone recogniser usually needs a phone-based hidden Markov model (HMM) and training involves annotated speech data. These supportive models are trained using the second type of training sets which are called *acoustic-model training sets*.

A schematic overview of the learning framework studied in this paper can be found in Fig. 1. Here, the processing takes places from bottom to top, and the multiple directed arrows indicate various combinations of processing steps. First, the spectro-temporal features are extracted from the speech signal (Section 3.2), resulting in either Mel Frequency Cepstral Coefficients (MFCC, c.f. Section 3.2.2) or Mutual Information Discriminant Analysis features (MIDA, c.f. Section 3.2.3). The horizontal arrow leading to the MIDA features from the left indicates that for the creation of MIDA features, annotated speech material is needed. In the next step, the spectro-temporal features are converted into posteriorgrams (Section 3.3), either by using soft-VQ clustering (c.f. Section 3.3.1) or a phone recogniser (c.f. Section 3.3.2). As for the MIDA feature extraction, the horizontal arrow leading to the phone recogniser from the left indicates that for this approach, annotated speech material is needed. The horizontal arrow leading to the soft VQ procedure from the right indicates that speech material is needed to train the code books. Finally, the posteriorgrams are converted to utterance-level representations (Section 3.3.3) by using Histograms of Acoustic Co-occurrence (HAC) after which the NMF training takes place (Section 3.4). The horizontal arrow leading to the NMF training phase visualises the fact this training is supervised with grounding information.

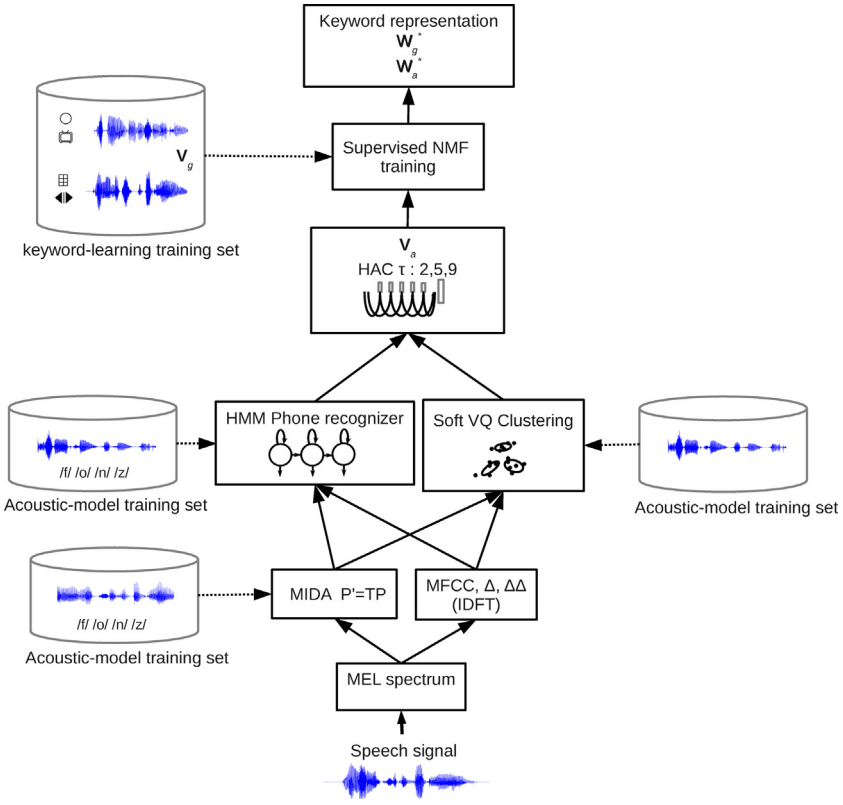

Fig. 1. Multi-layered architecture. The architecture allows to compose high-level acoustic units and facilitates associative learning between grounding signals and physical acoustic signals. The steps depicted at the same level are denoting exchangeable alternatives.

3.2. Feature extraction

3.2.1. MEL spectrum

Speech samples are transformed into Mel-spectra. We used a short time Fourier transform with the Hamming window of 25 ms and a frame shift of 10 ms, followed by a bank of Mel-scaled triangular filters. There are two alternatives in feature extraction resulting in either MFCC or MIDA features.

3.2.2. MFCC features

MFCC features (Davis and Mermelstein, 1980) are obtained by applying the Inverse Discrete Cosine Transform (IDCT) to the log-Mel spectral representation. The IDCT expresses the signal in terms of a sum of orthogonal cosine functions oscillating at different frequencies and their amplitudes correspond to the cepstral coefficients in MFCC features. The cepstra of the first 12 cosine functions ($c_0, \ldots, c_{11}$) and the log energy are retained. The 13-dimensional representation is augmented with their first and second order differences (Δ- and $\Delta\Delta$-features), yielding a total of 39 coefficients per frame. The MFCC features are mean and variance normalized per utterance.

3.2.3. MIDA features

Discriminant analysis algorithms for feature extraction are often based on a transformation which maximizes the between-class scatter and minimizes the within-class scatter. In Mutual Information Discriminant Analysis or MIDA (Demuynck, 2001), a linear transformation is sought that maximizes the mutual information between the transformed features and the target classes. The algorithm uses frame-based target class annotations. In our study, the target classes consist of phones, but other annotations can be used as well such as VQ clusters (see Driesen et al., 2012a). The MIDA transformed features, or MIDA features in brief, are a linear combination of 22 log-MEL spectral dimensions and their first and second order differences (Δ and $\Delta\Delta$). The MIDA features are ordered in terms of mutual information, and

we reduced the dimensionality to the 39 most informative MIDA features to conform with the dimensionality of the MFCC features.

3.3. Posteriorgram

The feature vectors obtained by the different feature extraction methods are transformed into a posteriorgram. A posteriorgram is a two dimensional data structure containing the posterior probabilities $\mathbf{P}_{t_i,\theta}$ that the observation in frame at time t_i with, $i = 1, \ldots, Q$ and Q the number of frames, originated from an acoustic unit $\theta \in \Phi$ with Φ the set of acoustic units. The dimensionality of a posteriorgram is $L \times Q$ with L the number of acoustic units.

We evaluate two alternative acoustic units: Soft VQ clusters (Driesen, 2012; Sun and Van hamme, 2011b) modelled by one Gaussian for each cluster in the feature space and phones modelled by a tri-state HMM. Both alternatives are usually based on models with parameters that are estimated beforehand.

3.3.1. Soft VQ posteriorgram

First, clusters are obtained by a codebook training procedure adopted from Driesen (2012) and Sun and Van hamme (2011b). The code book training procedure starts with one cluster joining all frames, and the cluster(s) are split iteratively in sub-clusters until the requested number of clusters is obtained. The iterations comprise two steps. In the first step, the cluster with the largest covariance is split into two clusters by replacing the centre of the respective cluster with two new centres located in the neighbourhood of the old one but shifted in opposite directions along the main axis of variation. In the second step, k-means clustering is applied using 15 iterations in which frames are partitioned into clusters based on the shortest distance to the centres, and then, centres are estimated for the new clustered frames. In the reported experiments, different code books were used of different sizes: $L = 20$, 100 and 400.

Secondly, all frames in the training set are partitioned in the obtained clusters and a full covariance Gaussian is estimated on all the frames that fall in each respective cluster.

During decoding, a posteriorgram for each frame is obtained by evaluating the probability density functions of the Gaussians at the location of the frame in the feature space and by normalizing the relative likelihoods of the Gaussians to one, i.e. representing each frame by a multinomial probability distribution where each entry denotes the chance that the frame-based observation was emitted by the respective Gaussian. We refer to the cluster-based posteriorgram with the term “soft VQ representation”.

3.3.2. Phone posteriorgram

An alternative for soft VQ representations are phone posteriorgrams. First, a phone recogniser is built by training tri-state HMM mono-phone models based on a training set with speech data and phonetic transcriptions for a particular phone alphabet with a size of L phones. A mixture model with G diagonal-covariance tied Gaussians is used to model the observation probabilities of the phone states.

In the decoding phase, the acoustic models and HMM topologies are used to build a directed acyclic graph, building on the ten best Viterbi scores (Wessel et al., 2001) in which each arc represents a phone. For each time frame, we accumulate the scores of the arcs passing the frame for each respective phone and we calculate posterior probabilities by using the forward-backward algorithm (see Rabiner, 1989). By normalising the accumulated scores, we obtain posterior probabilities.

Note that we do not use phone n-gram relations, to avoid the risk that the posterior probabilities are influenced by the average co-occurrence patterns in the data instead of reflecting the instantaneous acoustic sounds as such.

3.3.3. Histogram of acoustic co-occurrence

The posteriorgram of an utterance has a variable length that depends on the number of frames in an utterance. However, fixed-length vectors are required to compose the data matrix for NMF. The aim of HAC (Driesen et al., 2012a; Van Segbroeck and Van hamme, 2009; Van hamme, 2008) is to build a fixed-length vector for each utterance by accumulating the probability of observing a phone or a VQ cluster pair (α, β) for all possible $L \times L$ pairs over two frames shifted τ frames away from each other. The probability of co-occurrence can be accumulated over the whole utterance for every possible phone or VQ cluster pair resulting in a fixed length vector of $F = L^2$ entries. For utterance

n having q sequential frames, the co-occurrence score for the phone or the VQ cluster pair (α, β) with $\alpha, \beta \in \Phi$, and Φ the phone or code book set, can be expressed as follows

$$\left[\mathbf{v}_n^{\tau}\right]_{(\alpha,\beta)} = \sum_{t_i=0}^{q-\tau} \mathbf{P}_{t_i,\alpha}\mathbf{P}_{t_i+\tau,\beta} \tag{1}$$

and $\forall t_i$, $i = 1, \ldots, Q$, $\sum_{\theta\in\Phi}\mathbf{P}_{t_i,\theta} = 1$.

An utterance is then represented by an accumulation of phone or soft VQ co-occurrence probabilities. In the current experiment we used different τ lags with $\tau = 2$, 5 or 9 (Driesen and Van hamme, 2011a; Driesen et al., 2012b). Each utterance is represented by a single fixed-length column vector $\mathbf{v}_{a,n}$,

$$\mathbf{v}_{a,n} = \begin{bmatrix} \mathbf{v}_n^{\tau=2} \\ \mathbf{v}_n^{\tau=5} \\ \mathbf{v}_n^{\tau=9} \end{bmatrix} \tag{2}$$

in which all combinations $(\alpha, \beta) \in \Phi \times \Phi$ are stacked over different time lags for one whole utterance. Additionally, we implemented different code books of different sizes L and the number of code books is denoted by the constant C (Ons et al., 2012; Driesen and Van hamme, 2012b). All co-occurrence scores corresponding to the C code books are stacked in one vector per utterance for the soft VQ representation. For the collection of N utterances in the training set, the acoustic representation is denoted by $\mathbf{V}_a = [\mathbf{v}_{a,1}\mathbf{v}_{a,2}\ldots\mathbf{v}_{a,n}]$ with $n = 1, \ldots, N$.

3.4. NMF learning

In supervised NMF learning (Driesen et al., 2012a; Van hamme, 2008), the acoustic representation $\mathbf{V}_a$ of the training set is augmented with grounding information $\mathbf{V}_g$. In $\mathbf{V}_g$, the presence of keywords in each utterance is indicated as follows: There is one row in $\mathbf{V}_g$ for each keyword label and its entries represent the number of times that the respective keyword was uttered. $\mathbf{V}_a$ is a $(F \times N)$ matrix with F the acoustic feature dimension and $\mathbf{V}_g$ is a $(K \times N)$ matrix with K the number of keywords. Non-negative Matrix Factorization will decompose the matrix $[\mathbf{V}_g{}^T\mathbf{V}_a{}^T]^T$ into the product of two low-rank matrices

$$\begin{bmatrix} \mathbf{V}_g \\ \mathbf{V}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} \tag{3}$$

The purpose of the supervised NMF learning is to find the latent acoustic representation for each property or action needed to control a device. The columns in $\mathbf{W}_a$ represent the latent structure, i.e. the recurring acoustic patterns of the columns in $\mathbf{V}_a$ co-occurring with the semantic tags (see Section 2) that are represented by label vectors in $\mathbf{V}_g$ similar to the keywords in (Driesen et al., 2012a; Van hamme, 2008). The columns in $\mathbf{H}$ indicate which patterns, thus columns in $\mathbf{W}$, are combined to approximate the columns in $\mathbf{V}$. When the total set of vocal commands contains K semantic items for which the system needs to learn one word, $\mathbf{W}$ should count K columns, but we add D extra columns in $\mathbf{W}$ to model filler words. Note that filler words can also model synonyms for the first K words on condition that the synonym are frequently spoken by the user. Another approach to learn synonyms is by detecting the use of a second word for a particular label vector after which a second column in $\mathbf{W}$ is introduced for this label vector. However, the treatment of synonyms is not pursued in this study. We refer to the learned words with the term "keywords".

The representation of the keywords in $(\mathbf{W}_a)$ can be found by minimizing the Kullback-Leibler divergence between both sides of Eq. (3),

$$(\mathbf{H}^*, \mathbf{W}_a^*, \mathbf{W}_g^*) = arg\min_{(\mathbf{H},\mathbf{W}_a,\mathbf{W}_g)} D_{KL}\left(\begin{bmatrix} \mathbf{V}_g \\ \mathbf{V}_a \end{bmatrix} || \begin{bmatrix} \mathbf{W}_g \\ \mathbf{W}_a \end{bmatrix} \mathbf{H}\right) \tag{4}$$

Iterative update rules for minimizing a distance measure between the left and the right handside can be found in Driesen et al. (2012a), Lee and Seung (1999) and Van hamme (2008). Convergence is guaranteed towards a local optimum.

Keyword recognition can be tested on a test set. We denote the data matrix $\mathbf{V}$ and $\mathbf{H}$ for the factorization of the test set by $\mathbf{V}_r$ and $\mathbf{H}_r$. $\mathbf{H}_r$ is found by minimizing the Kullback-Leibler divergence between $\mathbf{V}_r$ and $(\mathbf{W}_a^*\mathbf{H}_r)$

$$\mathbf{H}_r^* = arg\min_{\mathbf{H}_r} D_{KL}(\mathbf{V}_r||\mathbf{W}_a^*\mathbf{H}_r) \tag{5}$$

$\mathbf{W}_a^*$ consists of the latent structure found in the training phase, one column vector for each word, and $\mathbf{H}_r$ indicates which words need to combine to approximate the utterance-based data of the representations of the test set $\mathbf{V}_r$. An optimal solution for $\mathbf{H}_r$ is guaranteed since Eq. (5) is a convex problem. The obtained matrix $\mathbf{H}_r^*$ is used to provide the keyword activation matrix $\mathbf{A}$,

$$\mathbf{A} = \mathbf{W}_g^*\mathbf{H}_r^* \tag{6}$$

$\mathbf{A}$ is a $(K \times N)$ matrix and each column in $\mathbf{A}$ corresponds to the respective column in $\mathbf{V}_t$. The higher the score in the rows of $\mathbf{A}$, the more activation for the respective keyword in the spoken test utterances.

4. Experiments

4.1. Overview

The goal of the experiments is to investigate the different processing flows to evaluate keyword recognition accuracy in the initial and final phase of the learning curve, that is, the curve representing the acquisition of words in function of the number of learning examples for the average user. Usually, the rate of learning is sharpest in the beginning and gradually evens out. In the reported experiments, we show stepwise improvements of the different proposed flows in the architecture demonstrated in Section 3.

In the first experiment (Section 4.3), we set a baseline by using soft VQ mid-level representations (see Section 3.3.1) and we introduce phone posteriorgrams as a substitute for the existing method of soft VQ representations. Phone posteriorgrams have been used in NMF for the discovery of latent phone patterns (Stouten et al., 2008), but to the best of our knowledge, it has not been used in NMF for the purpose of fast learning. In the second experiment (Section 4.4), we investigate the difference in performance using MFCC or MIDA features.

Contrary to the first two experiments, where the training material consisted of speech from different speakers, the training material in the third experiment is speaker-dependent. Speaker-dependent keyword training is pursued (Section 4.5) since the VUI is a personalized system. We refer to the third experiment as *user-centred keyword learning*.

Contrary to the first two experiments, speaker-dependent keyword training is pursued in the third experiment (Section 4.5) since the VUI is a personalized system and since it is one of the key characteristics of the VUI. We refer to it as *user-centred keyword learning*.

In the light of our aim to investigate the feasibility of using NMF in a self-learning VUI, it makes sense to distinguish *realistic* processing flows from *unrealistic* ones. A realistic processing flow is a simulation of the VUI training corresponding to a real-life VUI-user context allowing only available training data to train the supportive acoustic models. Realistic and unrealistic are closely related to speaker-dependent and speaker-independent models. Speaker-independent acoustic models using corpora such as those employed in the field of speech recognition are considered to be realistic since speaker-independent models can be trained beforehand in a lab. However, speaker-dependent supportive models are only considered realistic when the training set contains utterances spoken by the user, that is, utterances contained in the keyword-learning training set. However, since we are interested in the feasibility of using NMF in a personalised VUI system, it is relevant to verify NMF learning using acoustic-model training sets for which training is speaker-dependent and optimal. Such a processing flow is using unavailable data like for instance all utterances of one speaker in the ACORNS corpus. It is regarded as an unrealistic processing flow, but it serves as an upper bound for what NMF can achieve when supportive models are optimally trained. In Sections 4.6 and 4.7 we progress to more realistic speaker-dependent models. In the fourth experiments (Section 4.6), we focus on speaker-dependent code-book training. Since the speech of the user is scarce in the beginning of the VUI usage, the speaker-independent models trained on large corpora might outperform the speaker-dependent models. Because speaker-dependency refers to a realistic setting of the VUI usage, we refer to it as *user-centred codebook training*.

It is difficult to obtain good performances for both indicators, i.e., fast learning and high asymptotic accuracy, when using one processing flow. In Section 4.7, we combine two processing flows that are complementary in performing

well on both performance indicators in the previous experiments and we evaluate whether the combined processing flow is able to improve fast learning and high asymptotic accuracy.

4.2. Experimental setup

4.2.1. Databases and datasets

Data from different corpora is used to compose the training sets. All corpora contain normal speech. We use the data of the "Acquisition of Communication and Recognition Skills projects", *ACORNS* (Boves et al., 2007), to represent the commands that the end user utters to train the VUI. More particularly, we use the UK English subset of the corpus developed in the second year of the ACORNS project (Altosaar et al., 2010). The subset of the corpus consists of 13160 utterances produced by 10 different speakers: four speakers produced 2396 utterances and six speakers produced 596 utterances. Only the speech data of the first four speakers is selected since the amount of speaker-specific data is important for simulating the asymptotic behaviour of the VUI. Utterances consist of 1–4 different keywords embedded in a carrier sentence with unrelated filler words. In total, there are 50 unique predefined keywords and 30 filler words. The choice for the corpus fits well for the purpose of evaluating the learning curve of the VUI as the size and complexity of the data is similar to a common home automation task. We refer to the ACORNS subsets as the *keyword-learning training sets*.

We investigate fast learning by using keyword-training sets of increasing sizes, and we refer to the *series* of gradual increasing training sets with the term *fold*. In each fold the smaller sets are forming (nested) subsets of the larger training sets, i.e. representing snapshots of the same learning curve. The obtained accuracies for small training sets correspond to the accuracies that can be expected in the beginning of the learning curve when the VUI is put into service and the user starts the training. The accuracy of the largest training set corresponds to the accuracy for the case that the user has trained the VUI during a longer period of time. When NMF learning includes multiple speakers, the data is pooled for the different speakers and the mixed training sets count N utterances of all users with $N = 50$, 100, 200, 400, 800, 1600, 3200 and 7156 after excluding 32 utterances due to bad quality. The corresponding test sets count 2382 utterances after excluding 14 utterances. When NMF learning is user-centred, the folds are composed of training sets with training data from individual speakers and set sizes $N = 50$, 100, 200, 400, 800 and $N = 1790$, 1786, 1789 or 1791 for the largest training set of the fold depending on the respective speaker. The sizes of the corresponding test sets are 593, 594, 596 and 599 utterances for the four speakers, respectively. The average keyword occurrence for all 50 keywords is 3.0, 5.9, 11.8, 23.6, 47.3 and 105.7 times over all folds for data set sizes $N = 50$, 100, 200, 400, 800 and $N > 1785$, respectively.

We created three different folds with gradually increasing training set sizes for each processing flow under investigation by selecting utterances randomly without replacement. For each fold, **H** and **W** are estimated five times (see Section 3.4) using a different initialization (see Section 4.2.4) leading to different solution for the same fold. Initializing **H** and **W** five times for three folds results in 15 learning curves for each processing flow.

We use three different corpora to train the acoustic models at different layers: (1) ACORNS, (2) the "Wall Street Journal corpus recorded at the University of Cambridge, phase 0", *WSJCAM0* (Robinson et al., 1995), which is the UK English equivalent of a subset of the US English Wall street Journal corpus (WSJ0) and (3) the "Corpus Gesproken Nederlands", *CGN* (Oostdijk, 2000), which is a Dutch corpus consisting of continuous speech covering news bulletins selected from Dutch television and radio. ACORNS is a UK English corpus used here to simulate the spoken phrases of the VUI user, so the native language of the user in the training simulation of the VUI is implicitly set to UK English.

4.2.2. Feature dimensionality

The phone posteriorgrams (see Section 3.3.2) have dimension $L = 41$, 46 or 50 depending on the size of phonetic alphabets used in the transcriptions of ACORNS, WSJCAM0 or CGN, respectively. The phonetic alphabets also include one noise unit and one silence unit in addition to the phones. The noise and silence units model the silence and the non speech sounds such as coughts or breathing sounds. The phone models are trained using the open source software SPRAAK (Demuynck et al., 2008). For WSJCAM0, a mixture model with $G = 16822$ diagonal-covariance tied Gaussians is used to model the observation probabilities of the phone states. Similarly, $G = 5813$ and $G = 48{,}845$ tied Gaussians are used for ACORNS and CGN respectively. Phone posteriorgrams are converted to HAC's (see Section 3.3.3). The dimension of the HAC feature vector F depends on L and on the number T with T the number of frame-lags τ, $F = T \times (L^2)$ or $F = 5043$, 6348 and 7500 features depending on the training set of the phone recogniser.

The number of code books C is 3 and the dimension L for the code books was freely chosen with $L = 20$, 100 or 400 (see Section 3.3.1). Co-occurrence scores for soft VQ for three code books jointly and three frame lags produce fixed-length vectors in $\mathbf{V}_a$ with size $F = 3 \times (20^2 + 100^2 + 400^2) = 511200$ features.

The main portion of the probability mass per frame seems to originate from only a few phones or VQ clusters. We found out that only non-significant gains are obtained by taking more than the three largest probabilities into account for the soft VQ representations and more than the 10 largest probabilities for the phone posteriorgrams. Therefore, we only take the three highest probabilities per frame into account for soft VQ and the ten highest probabilities for phone posteriorgrams leading to $3 \times T \times C = 27$ and $10 \times T = 30$ non-zero entries per column in the posteriorgram for soft VQ and phones, respectively. Phone posteriorgrams usually lead to less sparse NMF problems but their dimensionality is considerably lower.

4.2.3. Naming

The successive steps of the processing flows from the bottom to the top of Fig. 1 are reflected in the names of the different graphs in each experiment. There are two possible acoustic-model training sets, one for learning the MIDA transformation and one for training the VQ Gaussians or the phone HMM. The processing flows using MFCC features start with "MFCC" and the processing flows using MIDA features start with "MIDA" followed by the training corpus for the MIDA transformation between parentheses: "ACN" for ACORNS, "WSJ" for WSJCAM0 or "CGN" for CGN. The second part of the names refers to the mid-layer representation "SVQ" for soft VQ and "PHN" for phone posteriorgrams, followed by the name of the training corpus used to train the code books or the phone HMM. An overview is presented in Table A1 in the Appendix A. The NMF procedure and the keyword-learning training sets are identical for the first two experiments. For the remaining three experiments, the keyword training sets are limited to the speech of one speaker. Since keyword learning is identical within each experiment, it is not incorporated in the names of the graphs. In the last two experiments, if the data of each individual speaker in ACORNS is also used to train codebooks, then, we added SD for speaker-dependent and SDD for speaker and set-size dependent inside the parentheses. SD means that the training material only consists of utterances from the respective speaker and SDD means that the training material is SD and limited to the keyword-learning training set.

4.2.4. NMF initialization

$\mathbf{H}_{init}$ and $\mathbf{W}_{init}$ denote the initialisation of $\mathbf{H}$ and $\mathbf{W}$, respectively

$$\mathbf{H}_{init} = \begin{bmatrix} \mathbf{V}_g + \lambda \mathbf{A}(K \times N) \\ \mathbf{B}(D \times N) + \gamma \mathbf{1}(D \times N) \end{bmatrix} \tag{7}$$

$$\mathbf{W}_{init} = \begin{bmatrix} \mathbf{I}(K \times K) + \lambda \mathbf{O}(K \times K) & \mathbf{P}(K \times D) + \theta \mathbf{1}(K \times D) \\ \mathbf{Q}(F \times (D + K))) \end{bmatrix} \tag{8}$$

with $K = 50$ and $D = 25$ and with $\lambda = 1e^{-4}$, $\gamma = 0.1$ and $\theta = 0.2$. All entries in $\mathbf{A}$, $\mathbf{B}$, $\mathbf{O}$, $\mathbf{P}$ and $\mathbf{Q}$ are i.i.d samples from the uniform distribution $\mathcal{U}(0, 1)$ with boundaries (0, 1). $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector with all ones. Note that keeping $\mathbf{W}_g$ set to identity is suboptimal, since tuned label weights in $\mathbf{W}_g$ are helpful to model the duration of spoken keywords or to model word parts over multiple columns that combine to one keyword (see Ons et al., 2013a). 100 iterations were used to find $\mathbf{H}$ and $\mathbf{W}$. Before each iteration the columns of $\mathbf{W}$ were normalised to sum to one. The parameters values were adopted from Driesen (2012).

4.3. Phone HMM versus soft VQ Gaussians

4.3.1. Introduction

In the first experiment, we compare two processing flows that differ only in the mid-layer representation soft VQ (Section 3.3.1) or phone posteriorgrams (Section 3.3.2).

Code books have been used before in NMF learning in numerous studies (Driesen et al., 2009, 2012a,b; Driesen and Van hamme, 2011b; Ons et al., 2012; Sun and Van hamme, 2011, 2011b, 2012) and we set the baseline by adopting multiple parameter settings from these studies. As in Ons et al. (2012), we used code books with different scales of granularity, $L = 20$, 100 and 400. Similar to Driesen et al. (2012a), we used HAC's with frame-lags, $\tau = 2$, 5 and 9.

Also the k-means procedure explained in Section 3.3.1 was shared with former studies (Driesen et al., 2009, 2012a,b; Driesen and Van hamme, 2011b; Ons et al., 2012; Sun and Van hamme, 2011, 2011b, 2012). However, the Gaussians and the soft VQ clusters were usually estimated by using the training data from the keyword-learning training set. Within the context of our study, it is regarded as an unrealistic simulation of the VUI-user usage (see Section 4.1). Nevertheless, this processing scheme is used to set a baseline in respect with earlier studies, and in the mean time, it serves as an upper bound for the best performance expected from soft VQ since any mismatch between code-book train- and test-set would result in less effective clusters degrading overall performance. The processing flow referring to this baseline setting is named MFCC_SVQ(ACN).

Code book training is completely unsupervised and data-driven, but the training of a phone HMM requires transcribed speech data. Spoken utterances of the user lack phonetic transcriptions, therefore, in a realistic VUI-user training scenario, phone models should be developed beforehand on transcribed speech data. We used WSJCAM0 for simulating the case that the phone models are trained on the native language of the end user, and CGN for the case that the trained phone models are originating from a different language. We refer to these processing flows with the names MFCC_PHN(WSJ) and MFCC_PHN(CGN), respectively. These two processing are considered realistic. Phone models trained on ACORNS are not considered realistic because the ACORNS speech data represents the spoken utterances of the user in our simulated VUI-user context. However, similar to MFCC_SVQ(ACN), it is still an interesting case for investigating the potential gain if phone models could be trained unsupervised. The processing scheme is called MFCC_PHN(ACN).

4.3.2. *Results and discussion*

In Fig. 2, the average accuracies and standard error bars of the learning curves are depicted for each processing flow. For the baseline experiment MFCC_SVQ(ACN), a score of 98.5% is obtained for the largest training set (see Table 2). It is a score comparable to 98.1% obtained in similar conditions and presented in Table 4.5, page 97 in Driesen (2012).

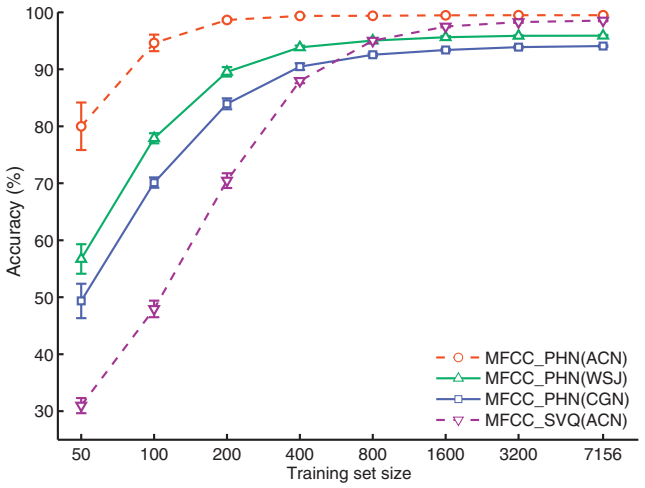

Fig. 2. Learning curves for processing flows using phone posteriorgrams or soft VQ as mid-level representation. Accuracy is plotted against the keyword-learning training set size. The error bars denote the standard error for the average accuracy over all folds and initializations.

Table 2
Accuracies plotted in Fig. 2 for keyword-learning training set sizes $N = 50$, 200 and 7156.

Training set size	50	200	7156
MFCC_PHN(ACN)	80.0	98.7	99.5
MFCC_PHN(WSJ)	56.7	89.5	95.9
MFCC_PHN(CGN)	49.3	83.9	94.1
MFCC_SVQ(ACN)	31.0	70.5	98.5

There, only one code book with $L = 500$ was used instead of multiple code books here, and the data was pooled over all 10 speakers instead of four speakers.

When we compare soft VQ against phone HMM, it is shown that the processing flows using phone posteriorgrams have higher accuracies in the beginning of the learning curve compared to the baseline. However, the flows MFCC_PHN(WSJ) and MFCC_PHN(CGN) level off earlier and accuracies are lower at the end. In Table 2, asymptotic accuracies of 95.9% and 94.1% are presented for the respective learning curves. Asymptotic accuracies are lower because the acoustic models are trained on speech from different corpora to conform with a realistic VUI training scenario, while the other two curves are using the user's speech to train the acoustic models in our simulation, i.e. data that is unavailable in a real training scenario.

When we compare the two flows having identical training sets in Table 2, that is MFCC_PHN(ACN) and MFCC_SVQ(ACN) (the dashed lines in Fig. 2), the Error Rate ($ER = 100\%$-accuracy) for $N = 50$ is 3.45 times lower for MFCC_PHN(ACN). A similar relative improvement with a factor of 2.86 is found between the same flows for $N = 7156$. Clearly, the choice of intermediate representations has a large influence on the learning speed.

4.4. MIDA features versus MFCC features

4.4.1. Introduction

In the low-layer representation of the architecture (see Fig. 1), two alternative spectro-temporal processing steps were explained. In the previous experiment, we used MFCC features in all processing flows, but here, we evaluate potential gains obtained from MIDA features in the same processing flows investigated before. We split the experiment in two parts: firstly, we evaluate the gains for MIDA using soft VQ, and secondly, we investigate MIDA features for the three phone recognisers adopted from the previous experiment.

In the first part, the three training sets for training the different MIDA transformations are ACORNS, WSJCAM0 and CGN. Code book training, the next step in the architecture, is then performed on the MIDA-transformed ACORNS features. According to the naming procedure (see Section 4.2.3), the processing flows are called: MIDA(ACN)_SVQ(ACN), MIDA(WSJ)_SVQ(ACN) and MIDA(CGN)_SVQ(ACN). By keeping the code book training set constant, the effects of the three MIDA-transformation are comparable.

In the second part of the experiment, we implement the MIDA variant for each phone recogniser treated in the previous experiment resulting in the following three processing flows MIDA(ACN)_PHN(ACN), MIDA(WSJ)_PHN(WSJ) and MIDA(CGN)_PHN(CGN).

4.4.2. Results and discussion

In Fig. 3(a) and Table 3 it is shown that all learning curves based on MIDA features have higher or equal accuracies than the ones based on MFCC features. However, most differences are non-significant.

In Fig. 3(b) and Table 3 it is shown that all MIDA variants of the phone recognisers depicted in Fig. 2 have higher accuracies over the whole range of the learning curves and some of these small differences are significant. The processing procedures investigated in the remaining experiments of this study are therefore all based on MIDA features.

The language of the training material influences the learning curves. When the supportive models are trained on a Dutch corpus (CGN), the scores are lower than when the models are trained on a British English corpus (WSJCAM). The best performance is obtained by using the same corpus for training both the acoustic and the keywords models, i.e. MIDA(ACN)_SVQ(ACN) and MIDA(ACN)_PHN(ACN).

4.5. User-centred keyword learning

4.5.1. Introduction

Contrary to the previous experiments (see Sections 4.3 and 4.4), user-centred NMF for keyword learning is pursued here with separate NMF keyword representations for every individual speaker instead of one keyword model counting for all four speakers together. Such a setup corresponds better to a realistic training context of the VUI where only a single end user is expected to train and use the system. We investigate the effect of speaker-specific input to the VUI on keyword learning. The investigated processing flows here adopt the MIDA variant of the processing flows investigated in the first experiment (Section 4.3). Note that the learning curves share the same names as some learning curves in

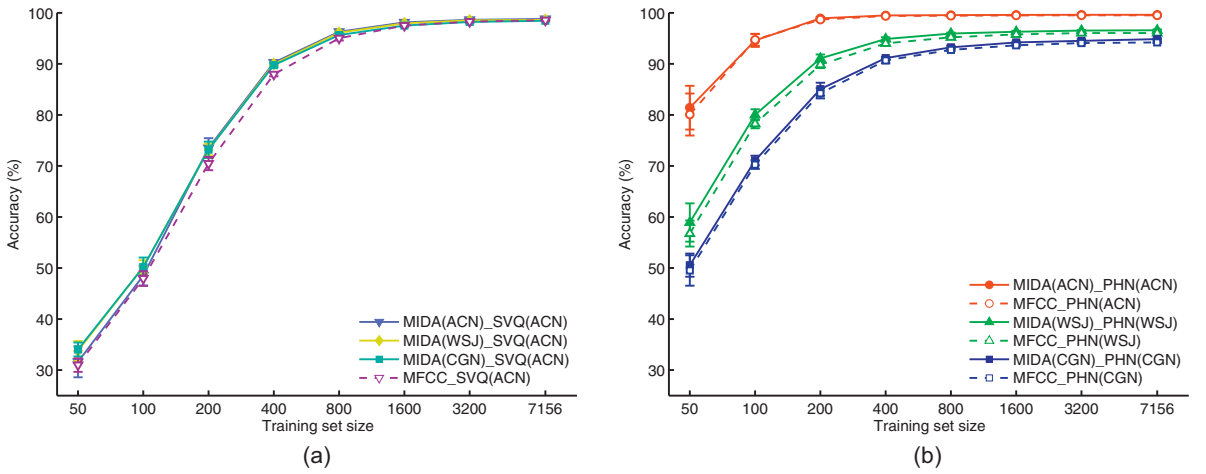

Fig. 3. Learning curves for processing flows comparing MIDA features against MFCC features for (a) soft VQ and (b) phone posteriorgrams. The error bars denote the standard error for the average accuracy.

Table 3
Accuracies plotted in Fig. 3 for keyword-learning training set sizes $N = 50$, 200 and 7156.

Training set size	50	200	7156
MIDA(ACN)_SVQ(ACN)	34.0	73.5	98.8
MIDA(WSJ)_SVQ(ACN)	33.7	73.2	98.6
MIDA(CGN)_SVQ(ACN)	31.6	73.2	98.5
MFCC_SVQ(ACN)	31.0	70.5	98.5
MIDA(ACN)_PHN(ACN)	81.4	98.9	99.6
MFCC_PHN(ACN)	80.1	98.7	99.5
MIDA(WSJ)_PHN(WSJ)	58.9	91.1	96.6
MFCC_PHN(WSJ)	56.8	89.8	96.0
MIDA(CGN)_PHN(CGN)	50.6	85.1	94.9
MFCC_PHN(CGN)	49.5	84.2	94.2

the preceding experiment because the low- and the mid-layers are identical, but the obtained accuracies might differ as keyword learning is user-dependent. The setup is identical to the previous experiments (see Section 4.2.1), except for the training sets containing utterances of each separate speaker only.

4.5.2. Results and discussion

The accuracies plotted in Fig. 4 represent the averaged score of all four speakers. The error bars reflect the standard deviation of the scores of the four speakers. The same qualitative differences as the ones observed in Fig. 2 can be observed in Fig. 4, but all accuracies are considerably higher. For instance, similar to Fig. 2, phone posteriorgrams outperform the soft VQ representation in the beginning of the learning curve, but the soft VQ representation outperforms two out of the tree streams based on phone posteriorgrams at the end of the learning curve. Exact accuracies are given in Table 4.

Accuracies are higher compared to speaker-pooled keyword learning because NMF models only need to take into account the vocalizations of a single speaker instead of all four speakers together. Discriminative representations are easier to build when the words are spoken by a single user because the words are spoken more consistently. Despite the fact that the largest training set is four times larger for the speaker-pooled folds in the preceding experiments compared to the folds here, the shorter learning curves here finish with higher accuracies. The highest accuracy, here, for all curves based on a realistic VUI-user learning context, is 98.9% for MIDA(WSJ)_PHN(WSJ) (see Fig. 4), but 96.6% for the same flow in the previous experiment (see Fig. 3(b)).

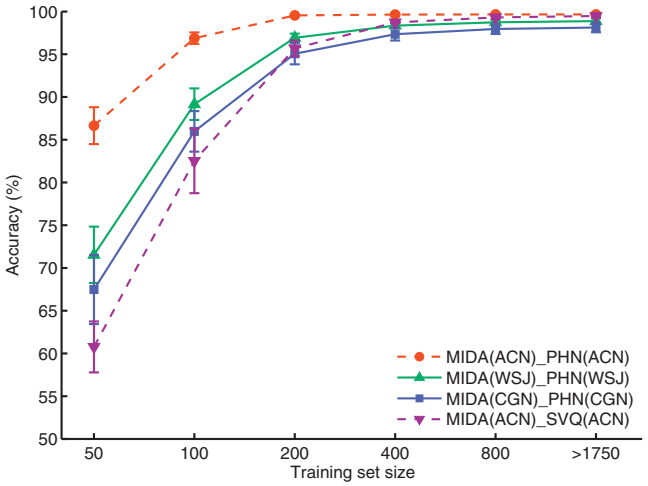

Fig. 4. Learning curves for user-centred keyword learning. The error bars denote the standard error for the mean accuracy of the four speakers.

Table 4
Accuracies plotted in Fig. 4 for keyword-learning training set sizes $N = 50$, 200 and >1750.

Training set size	50	200	>1750
MIDA(ACN)_PHN(ACN)	86.7	99.5	99.7
MIDA(WSJ)_PHN(WSJ)	71.5	96.9	98.9
MIDA(CGN)_PHN(CGN)	67.5	95.1	98.1
MIDA(ACN)_SVQ(ACN)	60.8	95.6	99.5

4.6. User-centred code book training

4.6.1. Introduction

The advantage of training code books beforehand is that large speech corpora can be used, such as those employed in the field of speech recognition. However, the acoustic-model training set is then recorded in different conditions (e.g. different microphones, different room acoustics and maybe cleaner speech) and with different speakers than the speech data originating from the user. We use WSJCAM0 as acoustic-model training set to simulate the case where the acoustic-model training set is different from the keyword-training set ACORNS. We refer to this processing flow with the name MIDA(WSJ)_SVQ(WSJ).

The speech data of the user has no phonetic transcriptions in a real VUI-usage environment, but, since code book training is data-driven, the user data can be used to train the code books. However, the data will be limited to the set of utterances that the user has spoken up until a particular moment in time, and thus, the training data is rather scarce especially during the initial VUI usage. We refer to the processing flow as MIDA(WSJ)_SVQ(ACN,SSD). We follow the code book training procedure explained in Section 3.3.1 but we add one constraint by prohibiting further splitting of clusters when the number of frames joining one cluster becomes less than 78 frames, a measure that allows for a more reliable estimation of the covariance matrix of the Gaussians. However, by fulfilling this constraint, the number of clusters is variable and gradually increases with the number of utterances in the training sets. For instance, for the training set sizes with $N = 50$, 100, 200, 800, and >1750, we obtained on average code book sizes of $L = 51$, 93, 148, 191 and 330 for all the folds.

The aim of this experiment is to investigate whether code book training for scarcely available but speaker-dependent data yields higher accuracies compared to the case where data is speaker-independent, but abundantly available in the field of speech recognition. These two realistic cases are accompanied by one unrealistic cases where code books are trained on all available speaker-dependent data in ACORNS. It serves as a reference for the case that large amounts of speech data from the user would be available before the VUI usage. We call the learning curve MIDA(WSJ)_SVQ(ACN,SD). Note that such a scenario can be realistic when speech from the end user is recorded beforehand for example by reading a standard text before the usage of the VUI. However, instructing a user to read a

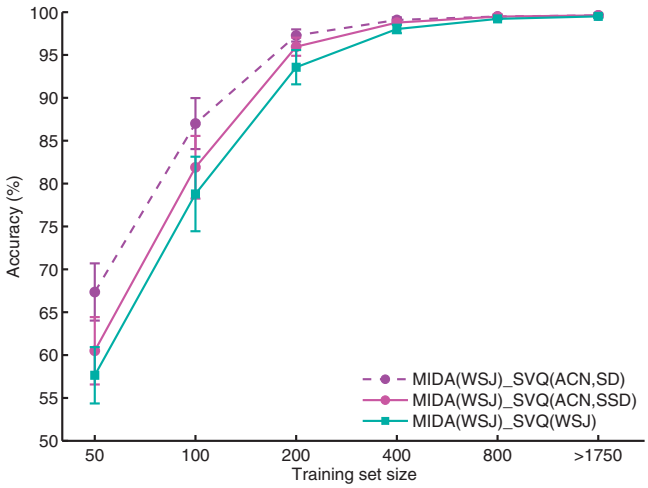

Fig. 5. Learning curves for user-centred keyword learning and speaker-(in)dependent code book training. The error bars denote the standard error for the mean accuracy of the four speakers.

standard text would require additional effort and the effect on the VUI depends strongly on the precise implementation of the instructions.

4.6.2. Results and discussion

When the two realistic learning curves are compared with each other (solid lines in Fig. 5), the best performances is obtained for MIDA(WSJ)_SVQ(ACN,SSD) over the whole range of the learning curve. Better performance was expected when plenty of speaker-dependent data is available allowing for better matching code books. However, small-sized code books matching the speaker's vocalizations also result in better scores in the beginning of the learning curve. In different words, small datasets matching the speaker's vocalization are preferable to many hours of speech data recorded in different conditions with different speakers and different vocabularies leading to more phonetic variation. User-centred soft VQ methods are also attractive for deviant speech for the reason that code books will give a good match to the end user and the training is unsupervised.

For the smallest training set size, the MIDA(WSJ)_SVQ(ACN,SSD) is 7% behind in absolute accuracy compared to the unrealistic best-case scenario MIDA(WSJ)_SVQ(ACN,SD) (see Table 5). The difference represents the potential gain that can be achieved hypothetically, if pre-recorded speech of the end user would be available beforehand.

4.7. Stream combination

4.7.1. Introduction

In this study, the goal is to combine the realistic processing flows that yielded the best results for the average user in all the former experiments. Within the set of realistic learning curves, MIDA(WSJ)_SVQ(ACN,SSD) provided the highest accuracy at the end of the learning curve and MIDA(WSJ)_PHN(WSJ) provided the highest accuracy in the beginning of the learning curve. By combining both processing flows, we investigate whether the best of both worlds can be obtained for the whole range of the learning curve.

The two flows are combined in NMF by stacking the data matrices $\mathbf{V}_a$ of both processing flows in one large data matrix giving both streams equal weights, i.e. both streams are normalised so the sum of all entries in each stream are equal. Naturally, weights can be tuned to favour one of the two performance indicators.

Table 5
Accuracies plotted in Fig. 5 for keyword-learning training set sizes $N = 50$, 200 and >1750.

Training set size	50	200	>1750
MIDA(WSJ)_SVQ(ACN,SD)	67.4	97.3	99.6
MIDA(WSJ)_SVQ(ACN,SSD)	60.5	95.8	99.6
MIDA(WSJ)_SVQ(WSJ)	57.6	93.6	99.5

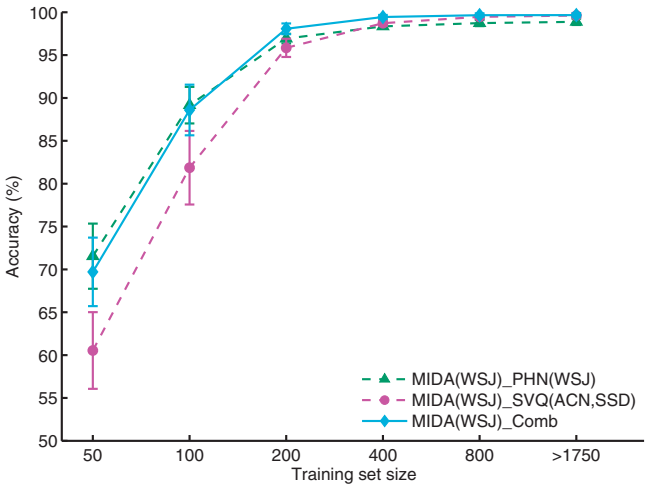

Fig. 6. Learning curves for the combination of two realistic processing flows adopted from the previous studies. The error bars denote the standard error for the mean accuracy of the four speakers.

Table 6
Accuracies plotted in Fig. 6 for keyword-learning training set sizes $N = 50$, 200 and >1750.

Training set size	50	200	>1750
MIDA(WSJ)_comb	69.7	98.0	99.7
MIDA(WSJ)_PHN(WSJ)	71.5	96.9	98.9
MIDA(WSJ)_SVQ(ACN,SSD)	60.5	95.8	99.6

The two streams MIDA(WSJ)_PHN(WSJ) and MIDA(WSJ)_SVQ(ACN,SSD) are adopted from the former two experiments in Sections 4.5 and 4.6. The combined stream is called MIDA(WSJ)_comb.

4.7.2. Results and discussion

When training set sizes are larger or equal to 200 examples, higher accuracies are obtained for MIDA(WSJ)_comb compared to its constituents: MIDA(WSJ)_PHN(WSJ) and MIDA(WSJ)_SVQ(ACN,SSD) (see Fig. 6 and Table 6). For training set sizes $N = 50$ and $N = 100$, the combined processing flow performs slightly worse than the best one of its constituent streams but still performs much better than the worst one of its constituent streams. For the largest training set size, the accuracies of the combined and the best of its constituent streams are similar. The combined processing scheme seems to approximates the best scores of its constituent counterparts and demonstrate therefore better overall performances.

5. General discussion

We investigated NMF-based VUI performance in a series of experiments simulating the realistic training conditions of the VUI-user context. For instance, the environmental local conditions of the user, like the room acoustics and the vocabulary spoken by the user are not known beforehand. Likewise, the data used for training phone HMM or MIDA transformations, like WSJCAM0 or CGN, are recorded in different conditions with different speakers and vocabularies than the data used for simulating the VUI training. We progressed to adapted models by first using user-centred NMF (Section 4.5) and secondly by using user-centred code book training (Section 4.6). Both steps improved the performance to a great extend.

We took more measures on the way to fast learning. In the first experiment (Section 4.3), we introduced phone posteriorgrams to enhance the feature vectors in NMF and we obtained better performance than the more common used soft VQ features (Driesen, 2012; Sun and Van hamme, 2011b). Also the use of MIDA features in the second experiment (Section 4.4) allowed for a slight improvement in performance.

The optimal performance is obtained by combining a phone recogniser trained on WSJCAM0 initiating a head start and the use of user-centred speaker and set-size dependent code book training allowing for high asymptotic accuracies

of the learning curve at the end (see Section 4.7). Both processing streams are considered realistic scenarios in the VUI usage context.

The user group consists of people with limb impairments for which voice control contributes to their independence of living. The majority of the user group is expected to have normal intelligibility, but some physical impairments are caused by neuromuscular diseases, therefore disarthric speech is expected too. The combined stream demonstrates promising results. One stream using phone posteriorgrams allows fast word learning for normal spoken utterances and one stream based on more basal soft VQ features allows to build up new word representations from scratch. The second stream is particularly interesting for people with a speech impairment. In future research, we will investigate whether the vocal user interface is able to anticipate dysarthric speech. Some preliminary research in that respect has been carried out in Ons et al. (2013b).

5.1. Posteriorgrams as feature vectors

Developmental studies demonstrate that humans build an intermediate representation of speech sounds in function of semantic content (Miyawaki et al., 1975; Werker and Lalonde, 1988). We show that machine learning of the semantic content of signals is largely improved when a mid-level representation is built based on speech-sound categories like phones or clusters. Especially, the use of posteriorgrams to enhance feature vectors seems to be a promising procedure in NMF learning. For instance, we used hard VQ in (Ons et al., 2012) for the baseline, MFCC_SVQ(ACORNS), and obtained a score of 32.6%, 48.2% and 95.6% for training sets in ACORNS of size 100, 200 and 9821 utterances. Here we obtained 47.9%, 70.5% and 98.5% for exactly the same conditions using the posteriorgram version of hard VQ, namely soft VQ.

The processing flows based on the posteriorgram of a pretrained phone recogniser are especially efficient in the beginning of the learning curve. Phones were modelled by a tri-state HMM expressing phones as variable sequences of frame-based acoustic observations. The generative HMM models can cope with many forms of spectro-temporal variation and in that sense, their structure incorporates a great deal of information on human speech in general by extracting information from large annotated corpora beforehand. In the mean time, they consist of very compact feature representations of the data at hand, facilitating the search of latent recurrent keyword patterns and allowing for fast word learning rates in NMF.

Conversely, the Gaussian models used in soft VQ are less complex, therefore limiting the training data required to estimate the parameters. Since a sufficient number of code words are required to accurately represent speech for recognition purposes, feature vectors based on soft VQ are less compact and more training examples for NMF keyword learning are required. However, code book training is data-driven, affording the pursuit of code books on-the-fly, leading to representative clusters regarding the speech of the user. The more user-specific clusters allow for better performance in the long run, especially for users with deviant or dysarthric speech. The positive results of the user-centred approach in training demonstrates the potential asset by grounding the learning process in the environment of the user.

Future research entails the evaluation of simple data-driven phone-like subword models (Driesen and Van hamme, 2012a; Sun et al., 2013) embodying the best of both worlds: user-centred acoustic models similar to soft VQ and generative HMM models allowing a compact feature representation. The major challenge is to predispose the acoustic models with a limited set of discriminative parameters: a limited set in order to achieve fast learning, but discriminative in order to obtain high accuracies in the long run. An alternative is to create a model with a growing number of parameters evolving to a more and more complex model as more data becomes available. Finally, the third possibility and the one pursued here is to combine different procedures with different strengths. In the last experiment (Section 4.7) we combined two processing flows at the front-end of NMF using equal weights for both streams. Future research entails finding optimal weights for different input streams based on the work of Driesen and Van hamme (2012b) and the dynamical adaptation of weights in function of the learning curve. Since Phone posteriorgrams and soft VQ level off at different instants, it is likely that optimal weights will change during the learning process.

5.2. Related work on fast learning

The exact experimental evaluation of our results with others is out of the scope of this study as it is difficult to compare results when they are based on different databases, procedures and scoring. For instance, we tested the feasibility of our approach and limited ourselves to a realistic learning scenario. The performance indicator accuracy is not complementary to the more common used WER in the sense that the accuracy does not comprise correct keyword

order in an utterance but only the proportion of correctly detected keywords in utterances with one to four keywords embedded in it. On the other hand, the categorical complexity of our task consists of 50 keywords and 30 filler words while a database, like for instance TIDIGITS, contains a lexicon of eleven words and a corpus like WSJCAM0 contains a lexicon of 64,000 words. Moreover, the supervision in our task is weak in the sense that it does not comprise word order or segmentation. Therefore, the comparison of our approach with related work is rather qualitative.

Fast vocabulary acquisition has also been investigated in an HMM architecture. In Clemente et al. (2012), the lexicon (9 digits, "oh" and "zero") from the TIDIGITS database was learned from just a few training examples with supervision. In their framework, optimal parameters were first sought for the initialization of the HMM by a multiple sequence alignment procedure in which an initial ergodic HMM was transformed into multiple left-right HMM's, one for each word. They used a large margin classifier to obtain good generalization to new instances as the classifier was trained on a few examples. For continuous speech, they obtained an average word error rate (WER) of 13.7% after three learning examples and 1.7% after 10 learning examples. The models of Clemente et al. (2012) work speaker-independent and their learning procedure is incremental.

Fast learning in an NMF framework has been investigated by ten Bosch et al. (2009). They pursued a computational model for the discovery of new words by young infants. In the task at hand, a vocabulary of 13 keywords embedded in a carrier sentence was learned and it was claimed that 20–25 learning examples per word were needed to approximate a recognition accuracy at asymptotic level. The same NMF procedure and data subset was pursued in Driesen and Van hamme (2012a) to learn a new vocabulary of 10 extra keywords after the acquisition of a vocabulary of 40 keywords and some filler words. They used self-discovering HMM subword units to enhance the acoustic input and they achieved similar acquisition rates to the ones presented in this study. Their investigation was rather aimed at the adaptation capacity of a fully trained NMF model to newly encountered words. Our work builds further on former NMF studies. By using phone posteriorgrams and more user-centred acoustic and keyword models (Section 4.7), we obtained an average ER of 26% after three learning examples and an ER of 1.5% after 10 learning examples of a keyword. Our method has an advantage over other NMF-based approaches because we use a pretrained phone recogniser trained on annotated databases.

In Ons et al. (2012), it was found that accuracies mainly depend on the number of correct examples per keyword and not on the number of utterances in the training set. If an average command consists of two keywords, for instance an object name and an action, then 2.25 correct demonstrations per command is needed on average to obtain a keyword recognition rate above 90%. Five correct demonstrations on average allows to reach asymptotic levels. We think that the average user will experience this training effort as a reachable goal. In that sense, fast learning in a realistic setting – which was the aim of our study – is achieved for normal speech.

5.3. Conclusion

We aim at designing a VUI that learns to understand normal and ultimately deviant speech by associating spoken commands to actions on a device during its usage. The VUI is trained by the end user by mining the speech input and the changes that are provoked on a device. The real learning process will take place in the environment of the user but it is simulated in our experiments in a realistic manner as a machine learning problem grounded with keyword labels, i.e. labels that specify the action on a device. We focussed on fast learning and high asymptotic accuracy of the learning curve.

Simple commands consisting of two keywords, like "Switch on the lights, please", can be learned by five demonstrations. Fast learning in a realistic setting – which was the aim of our study – was therefore achieved and we demonstrated fast learning by taking several measures on the way: phone posteriorgrams were introduced in the first experiment, MIDA features were pursued in the second experiment and user-centred NMF for keyword learning improved performance in the third experiment. In the fourth experiment, the results were in favour of user-centred code books trained on scarce data instead of using massive amounts of data from different speakers. Finally, for the combined processing flow in the fifth experiment, we obtained an accuracy in keyword detection of 99.7% (starting from 98.5% for the baseline) and we improved the accuracy for the smallest training set from 30.9% using state-of-the-art NMF approaches to 69.7%, that is a reduction in error rate of more than a factor two. Additionally, we focussed on realistic training scenarios to have a sense on how such a system would perform in a real-life training scenario as grounding of the VUI training in the user's environment is the most important key-aspect of the self-learning VUI.

Appendix A. Overview processing streams

See Table A1.

Table A1
The naming convention for different processing flows with respect to the low- and mid-layer data preparation for NMF-based keyword learning. Only processing flows used in the experiments are depicted. Italic formatted names indicate processing flows which are regarded as unrealistic because they make use of unavailable user-specific data to train the acoustic models. "SD" refers to speaker-dependent training and "SDD" refers to speaker and set-size dependent training.

		Training corpus	Low-layer			
			MFCC features	MIDA features		
			No training	ACORNS	WSJCAM0	CGN
Mid-layer	Soft VQ	ACORNS	*MFCC_SVQ(ACN)*	*MIDA(ACN)_SVQ(ACN)*	*MIDA(WSJ)_SVQ(ACN)*	*MIDA(CGN)_SVQ(ACN)*
		WSJCAM0			MIDA(WSJ)_SVQ(WSJ)	
		CGN				
		ACORNS, SD			*MIDA(WSJ)_SVQ(ACN,SD)*	
		ACORNS, SSD			MIDA(WSJ)_SVQ(ACN,SSD)	
	Phones	ACORNS	*MFCC_PHN(ACN)*	*MIDA(ACN)_PHN(ACN)*		
		WSJCAM0	MFCC_PHN(WSJ)		MIDA(WSJ)_PHN(WSJ)	
		CGN	MFCC_PHN(CGN)			MIDA(CGN)_PHN(CGN)

References

Akata, Z., Thurau, C., Bauckhage, C., 2011. Non-negative matrix factorization in multimodality data for segmentation and label prediction. In: 16th Computer Vision Winter Workshop.

Altosaar, T., ten Bosch, L., Aimetti, G., Koniaris, C., Demuynck, K., van den Heuvel, H., 2010. A speech corpus for modeling language acquisition: Caregiver. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (Eds.), Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta.

ten Bosch, L., Driesen, J., Van hamme, H., Boves, L., 2009. On a computational model for language acquisition: modeling cross-speaker generalisation. In: Text, Speech and Dialogue. Springer, pp. 315–322.

Boves, L., ten Bosch, L., Moore, R., 2007. Acorns-towards computational modeling of communication and recognition skills. In: Proc. IEEE Int. Conf. on Cognitive Informatics, California, USA, pp. 349–355.

Caicedo, J.C., BenAbdallah, J., Gonz ález, F.A., Nasraoui, O., 2012. Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization. Neurocomputing 76, 50–60.

Clark, H.H., Schaefer, E.F., 1989. Contributing to discourse. Cogn. Sci. 13, 259–294.

Clemente, I.A., Heckmann, M., Wrede, B., 2012. Incremental word learning: efficient hmm initialization and large margin discriminative adaptation. Speech Commun. 54, 1029–1048.

Davis, S.B., Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust. Speech Signal Process. 28, 357–366.

Demuynck, K., 2001. Extracting, Modelling and Combining Information in Speech Recognition. K.U. Leuven, ESAT (Ph.D. thesis).

Driesen, J., 2012. Discovering Words in Speech Using Matrix Factorization. K.U. Leuven, ESAT (Ph.D. thesis).

Driesen, J., ten Bosch, L., Van hamme, H., 2009. Adaptive non-negative matrix factorization in a computational model of language acquisition. In: Proc. Interspeech, Brighton, UK, pp. 1711–1714.

Driesen, J., Gemmeke, J., Van hamme, H., 2012a. Weakly supervised keyword learning using sparse representations of speech. In: Proc. ICASSP, Kyoto, Japan, pp. 5145–5148.

Driesen, J., Gemmeke, J.F., Van hamme, H., 2012b. Data-driven speech representations for NMF-based word learning. In: Proceedings of the Workshop on Statistical and Perceptual Audition, Portland, OR, USA.

Driesen, J., Van hamme, H., 2011a. Modelling vocabulary acquisition, adaptation and generalization in infants using adaptive Bayesian PLSA. Neurocomputing 74, 1874–1882.

Driesen, J., Van hamme, H., 2011b. Modelling vocabulary acquisition, adaptation, and generalization in infants using adaptive bayesian plsa. Neurocomputing 74, 1874–1882.

Driesen, J., Van hamme, H., 2012a. Fast word acquisition in an NMF-based learning framework. In: Proc. ICASSP, Kyoto, Japan, pp. 5137–5140.

Driesen, J., Van hamme, H., 2012b. Supervised input space scaling for non-negative matrix factorization. Signal Process 92, 1864–1874.

Gemmeke, J., Ons, B., Tessema, M., Van de Loo, J., De Pauw, G., Daelemans, W., Huyghe, J., Derboven, J., Vuegen, L., Van Den Broeck, B., Van hamme, H., 2013. Self-taught assistive vocal interfaces: an overview of the aladin project. In: Proceedings of Interspeech.

Heinroth, T., Grotz, M., Nothdurft, F., Minker, W., 2012. Adaptive speech understanding for intuitive model-based spoken dialogues. In: Proc. LREC, pp. 1281–1288.

Demuynck, K., Roelens, J., Van Compernolle, D., Wambacq, P., 2008. Spraak: an open source speech recognition and automatic annotation kit. In: Proc. International Conference on Spoken Language Processing, Brisbane, Australia.

Kuhn, R., Junqua, J.-C., Nguyen, P., Niedzielski, N., 2000. Rapid speaker adaptation in eigenvoice space. IEEE Trans. Speech Audio Process. 8, 695–707.

Lee, D., Seung, H., 1999. Learning the parts of objects by nonnegative matrix factorization. Nature 401, 788–791.

van de Loo, J., Gemmeke, J.F., De Pauw, G., Driesen, J., Van hamme, H., Daelemans, W., 2012. Towards a self-learning assistive vocal interface: vocabulary and grammar learning. In: Proc. of the workshop Speech and Multimodal Interaction in Assistive Environments (SMIAE).

Miyawaki, K., Jenkins, J., Strange, W., Liberman, A., Verbrugge, R., Fujimura, O., 1975. An effect of linguistic experience: the discrimination of [r] and [l] by native speakers of Japanese and English. Atten. Percept. Psychophys. 18, 331–340.

Ons, B., Gemmeke, J.F., Van hamme, H., 2012. Label noise robustness and learning speed in a self-learning vocal user interface. In: Proc. of the International Workshop on Spoken Dialog Systems (IWSDS), Ermenonville, France.

Ons, B., Gemmeke, J.F., Van hamme, H., 2013a. NMF-based keyword learning from scarce data. In: Automatic Speech Recognition and Understanding Workshop, ASRU, Olomouc, Czech Republic.

Ons, B., Tessema, N., van de Loo, J., Gemmeke, J.F., 2013b. A self learning vocal interface for speech-impaired users. In: Proceedings SLPAT 2013, pp. 1–9.

Oostdijk, N., 2000. The spoken Dutch corpus. Overview and first evaluation. In: Proc. LREC, Genoa, Italy.

Paek, T., Chickering, D., 2007. Improving command and control speech recognition on mobile devices: using predictive user models for language modeling. User Model. User-Adap. Inter. 17, 93–117.

Parker, M., Cunningham, S., Enderby, P., Hawley, M., Green, P., 2006. Automatic speech recognition and training for severely dysarthric users of assistive technology: the stardust project. Clin. Linguist. Phon. 20, 149–156.

Potamianos, A., Narayanan, S., 1998. Spoken dialog systems for children. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 197–200, IEEE volume 1.

Quine, W., 1964. Word and Object, vol. 4. MIT Press.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77, 257–328.

Robinson, T., Fransen, J., Pye, D., Foote, J., Renals, S., 1995. Wsjcam0: A British English speech corpus for large vocabulary continuous speech recognition. In: Proc. ICASSP, Detroit, Michigan, USA.

Stouten, V., Demuynck, K., Van hamme, H., 2008. Discovering phone patterns in spoken utterances by non-negative matrix factorization. IEEE Signal Processing Letters 15, 131–133.

Sun, M., 2012. Constrained Non-negative Matrix Factorization for Vocabulary Acquisition from Continuous Speech. K.U. Leuven, ESAT (Ph.D. thesis).

Sun, M., Van hamme, H.,2011. Image pattern discovery by using the spatial closeness of visual code words. In: 18th IEEE International Conference on Image Processing (ICIP). IEEE, pp. 205–208.

Sun, M., Van hamme, H., 2011b. A two-layer non-negative matrix factorization model for vocabulary discovery. In: ICML'11 Symposium on Machine Learning in Speech and Language Processing, Bellevue, Washington, USA.

Sun, M., Van hamme, H.,2012. Tri-factorization learning of sub-word units with application to vocabulary acquisition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 5177–5180.

Sun, M., et al., 2013. Joint training of non-negative tucker decomposition and discrete density hidden Markov models. Comput. Speech Lang. 27, 969–988.

Van hamme, H., 2008. Hac-models: a novel approach to continuous speech recognition. In: Proc. Interspeech, Brisbane, Australia, pp. 255–258).

Van Segbroeck, M., Van hamme, H., 2009. Unsupervised learning of time-frequency patches as a noise-robust representation of speech. Speech Commun. 51, 1124–1138.

Werker, J., Lalonde, C., 1988. Cross-language speech perception: Initial capabilities and developmental change. Dev. Psychol. 24, 672.

Wessel, F., Schluter, R., Macherey, K., Ney, H., 2001. Confidence measures for large vocabulary continuous speech recognition. IEEE Trans. Speech Audio Process. 9, 288–298.

Bart Ons received the M.Sc. degree in Industrial Sciences from Groep T in Leuven in 1994 and the M.Sc. degree in psychology from the KU Leuven in 2003. In 2010, he received a Ph.D degree from the KU Leuven on the subject of visual perception and categorization of 2-D contour shapes. He is currently working on speech recognition in the department of Electrical Engineering. His main research interests are visual and auditory perception and recognition.

Jort Florent Gemmeke is a postdoctoral researcher at the KU Leuven, Belgium. He received the M.Sc. degree in physics from the Universiteit van Amsterdam (UvA) in 2005. In 2011, he received the Ph.D degree from the University of Nijmegen on the subject of noise robust ASR using missing data techniques. He is known for pioneering the field of exemplar-based noise robust ASR. His research interests include noise robust speech recognition, acoustic modeling, dysarthric speech recognition and audio event detection.

Hugo Van hamme received the PhD degree in electrical engineering from Vrije Universiteit Brussel (VUB) in 1992, the M.Sc. degree from Imperial College, London in 1988 and the Masters degree in engineering ("burgerlijk ingenieur") from VUB in 1987. From 1993 till 2002, he worked for L&H Speech Products and ScanSoft, initially as senior researcher and later as research manager. Since 2002, he is a professor at the department of electrical engineering of KU Leuven. His main research interests are: applications of speech technology in education and speech therapy, computational models for speech recognition and language acquisition and noise robust speech recognition.